(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

TOWARDS AN INFORMATION MODEL AND MECHANISMS FOR DESIGN RATIONALE CAPTURE AND USE

by

Debra A. Morris
and
Tina M. Paco

September 1993

Principal Advisor: Balasubramaniam Ramesh

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1993	3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE TOWARDS AN INFORMATION MODEL AND MECHANISMS FOR DESIGN RATIONALE CAPTURE AND USE		5. FUNDING NUMBERS
6. AUTHOR(S) Morris, Debra A. and Paco, Tina M.		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.		
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE A

13. ABSTRACT *(maximum 200 words)*

Every year the Department of Defense's (DoD) expenditures on software alone amount to almost ten billion dollars, with maintenance costs comprising the majority of this figure. Recent studies have indicated that an effective solution to help curtail the large maintenance cost is by capturing the rationale which was used to create the systems requirements and designs, and using this information throughout the life cycle. However, various models proposed by current research for capture of design rationale address only some specific aspects of the design process rather than the entire design process. This thesis identifies the important components of a comprehensive design rationale information model, proposes mechanisms to facilitate their capture, and identifies the generic functionalities of a design rationale management tool to use the rationale in various systems development activities.

14. SUBJECT TERMS Design rationale capture, Design reuse			15. NUMBER OF PAGES 89
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

TOWARDS AN INFORMATION MODEL AND MECHANISMS FOR DESIGN RATIONALE CAPTURE AND USE

by

Debra A. Morris
Lieutenant, United States Navy
B.S., University of Utah, 1985

Tina M. Paco
Lieutenant, United States Navy
B.S., Tulane University, 1985

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT
from the
NAVAL POSTGRADUATE SCHOOL
September 1993

Authors: Debra A. Morris Tina M. Paco

Approved by: Professor B. Ramesh, Principal Advisor

Kishore Sengupta

Professor ~~K. Sengupta, Associate~~ Advisor

Professor D. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

Every year the Department of Defense's (DoD) expenditures on software alone amount to almost ten billion dollars, with maintenance costs comprising the majority of this figure. Recent studies have indicated that an effective solution to help curtail the large maintenance cost is by capturing the rationale which was used to create the systems requirements and designs, and using this information throughout the life cycle. However, various models proposed by current research for capture of design rationale address only some specific aspects of the design process rather than the entire design process. This thesis identifies the important components of a comprehensive design rationale information model, proposes mechanisms to facilitate their capture, and identifies the generic functionalities of a design rationale management tool to use the rationale in various systems development activities.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

Every year the Department of Defense's (DoD) disbursements on software alone amount to almost ten billion dollars. Many sources have estimated that maintenance costs comprise seventy to eighty percent of that figure. With the shrinking defense budget, it becomes paramount to develop systems which are easier and more economical to maintain and implement (Endoso, 1992, p. 6).

Recent studies have recognized that effectively capturing and using rationale throughout the development life cycle will help decrease the rising costs of software maintenance (Dhar and Ramesh, 1992, pp. 498-499). In the later stages of the life cycle, requirements and design rationale can be useful in change management and can facilitate reuse of components. The DoD can also utilize such data as part of a comprehensive requirements traceability effort.

Design rationale are often the outcome of deliberations between members of the design team. The goal of capturing design deliberations is to enable the understanding of specifications, design components, or artifacts, and the decision making process that creates them. A historical

record of rationale identifying how the requirements and design evolved will provide the knowledge necessary to recognize repercussions of changing the requirements in the design and the final product.

Capture and maintenance of rationale becomes even more important in the context of shrinking DoD budgets, as systems are expected to have longer and longer life cycles. Rather than designing new systems increased attention is being given to modification of those already in existence. Such modification efforts usually encompass three basic areas: reengineering, reuse, and maintenance; all of which can be supported by the capture and use of design rationale.

The first step in any reengineering effort is the understanding of the initial design of the system. Having a method to identify the design rationale implemented in the original system will foster such an understanding. Reuse efforts will be greatly benefited by having information on not just the artifacts created, but also how they were created.

During maintenance efforts, involving hardware upgrades on or improvements to software, having the design rationale available may enable maintainers to make modifications without unduly affecting other aspects of the system.

Current models for the capture of design rationale are not comprehensive as they address only aspects of design rather

than the entire design process. In arriving at the basic components of any comprehensive design rationale information model, one must first have an understanding of design processes. Most large scale design efforts involve design teams rather than single designers working independently. To understand the design rationale which arise from group deliberations, an appreciation of group dynamics is essential. At the center of group dynamics is group communication. Therefore, a model to capture design rationale must also incorporate features which reflect aspects of group communications. Further, employment of the model also necessitates the exploration of mechanisms which could capture the information necessary to populate the model and facilitate the use.

B. OBJECTIVES

Our main goals in the research are the identification of the components of a rationale information model, mechanisms to facilitate their capture, and the generic functionalities of a design rationale management tool to use the rationale in various systems development activities.

C. SCOPE

Our original intent was to develop a prototype model for design rationale capture which could utilize current

technologies to support their capture and use. However, due to unavailability of hardware support, we refocused our goals. Our research began with a literature survey on current models of group work and group communication. We then investigated various models for representing design rationale. We proceeded to examine methods for effective capture of the design rationale. Literature on current technologies led us to focus on the area of Computer Supported Cooperative Work (CSCW). Specifically, we reviewed the applicability of mechanisms employing multimedia techniques for capturing design rationale.

Besides current literature, other sources that provided valuable information towards our findings include:

- Multimedia Expo, Santa Clara, California (October, 1992)
- meeting with K. Baudin and J. Givens at NASA Ames, Palo Alto, California, to discuss current design rationale tools such as Dedal (December, 1992)
- meeting V. Baya at Stanford's Center for Design Research, Palo Alto, California (December, 1992)
- presentation given by S. Hashim on the Issue Based Information System (IBIS) at the Naval Postgraduate School, Monterey, California (January, 1993)
- visit to Stanford's multimedia lab to include a demonstration of a tool called Maestro (February, 1993)
- Groupware'93 conference and exhibition, San Jose, California (August, 1993)

D. ORGANIZATION OF THE THESIS

Chapters II and III explore what we view as two basic areas that will facilitate the understanding of the design process: specifically, design and communications. Chapter IV will describe current models, methodologies, and tools which address the capture of design rationale. Finally, Chapter V will answer our research questions by presenting components of a design rationale information model and suggesting basic functionalities to support its capture and use. Chapter VI will provide our recommendations and conclusions.

II. DESIGN

A. DESIGN HISTORY

The design process is defined as "any activity that leads to the creation of artifacts" (Dhar and Ramesh, 1992, p. 498). Artifacts include any tangible output, such as documentation, graphical drawings and prototype products. Besides focusing on the creation of artifacts, the design process also produces information about the artifacts which are understandable and useful not only to the designers but also to those outside the original design team. The early stages of design incorporate conceptual ideas and formulate them into structured artifacts such as design specifications, prototypes, and graphic representation of data.

Different factors that influence the design process include project task, project complexity, designers' experience and design group size (Kellogg, Maass, and Rosson, 1988, pp. 1288-1298).

Project task is the component which describes the basic nature of the project, such as building an interface from scratch or redesigning a retrieval mechanism. The efforts involved in creating an artifact from the most basic beginnings may encompass a different type of activity in

contrast to modifying an artifact that already exists. Other influences that may impact the project composition involve the dynamics which surround the particular design task such as constrained or open-ended requirements.

Project complexity describes the magnitude of technical requirements of the project. For example, designing an operating system may be more complex than redesigning a database. Size of the project is an important consideration in this area also.

Each designer contributes his/her own individual and diverse knowledge into the process which must be collaborated within the group to produce any artifact. An individual with many years of experience working within a group may not articulate basic assumptions about design premises to other group members; whereas novice designers working together may find it necessary to articulate the most basic premises of the project. A group whose members vary in experience level may exhibit characteristics of both.

It is the collaboration of individuals' efforts in the group atmosphere that produces many of today's large scale systems. Each individual in the group is influenced by other group members. As the size of the group increases so does the complexity of group interactions.

Design is an evolutionary process where the output of each cycle is used in refining the input to the next cycle. One phase precedes another and all phases are interdependent. The simplest of changes to the early phases may create influences which have dramatic effects on the performance or size of the overall system. A medium for tracking these early changes and their resulting effect on the overall system would greatly facilitate the design process. However as "design emerges from a context of human desires and needs, subject to all the foibles of human activity," (Carroll, 1992, p. 4) clearly identifying all influences which lead to changes may be difficult.

Another dynamic in the design process is the incorporation of new knowledge into the designers' existing premises which influence design practices. Learning arises from comparing new information against a mental template that is formed by past experience; the design process likewise is refined by constructing new ideas from the application of existing concepts and models into designers' existing premises. The formulation of this new information is predominantly necessitated by changing requirements which have their origins outside the design team. This refinement is what is described as the iterative process of design.

As an intuitive process, we view design as not only intuitive, but conversely, judgmental as well as learned. Each member within design teams brings his/her personal experiences, learned skills, and expertise into the group. It is the combination of all the members' individual backgrounds which gives each group a unique set of preconceived notions or insights. These notions and insights are refined as interactions occur between group members, thus the group will formulate additional insights upon which to base judgements in the design realm.

> While conscious, rational thought and preparation may precede (and even be necessary for) such insight, the insight itself is not part of any conscious, formalizable process. It is the designer's intuition that pulls together the appropriate parts of his knowledge that 'have no affinity apart from...intuitive understanding' and drives the software input to output transformation (Carroll, 1992, p. 9).

Capturing the rationale underlying the design process may provide tangible insights into the formulation of group intuition.

When a decision is required in the design process, the designer could examine different alternatives. Each time an alternative is chosen or discarded a possibility exists to gain knowledge from recording such an interaction; information drawn from the recording could provide insight into

understanding and modifying the design process. Often, the reflection of the alternative chosen is clear in project artifacts; however there may not be sufficient documentation or tracking of alternatives which were not chosen. Information regarding those alternatives not chosen should also be maintained in an understandable format because they may become relevant with the addition of new or changing requirements.

B. DESIGN RATIONALE

Design rationale is the reasoning behind or decision making process involved in the creation of an artifact (Dhar and Ramesh, 1992, p. 498; Grueber and Russell, 1992, p. 111). Design rationale serves multiple purposes: definition of unstated assumptions, clarification of dependencies and constraints, and justification or validation of design decisions (Grueber and Russell, 1992, pp. 111-118). Greenbaum and Kyng fittingly described the purpose of design rationale when they stated the following:

> A user wanting to change a system will want to know what changes are already there and their history. This history includes the changes that have been made, who made them, and for what purpose, including the work practices that the user had in mind when modifying the system. All of this is needed for people to be able to continue to evolve the system coherently (Greenbaum and Kyng, 1991, p. 234).

Although the formulation of design rationale is an integral part of design, capturing design rationale is sometimes a time-consuming, secondary concern. A basic question to answer would be, "why even bother?" To think that design starts from scratch is an unrealistic assumption; with the incorporation of design rationale designers are able to prevent the phenomena know as "reinventing the wheel." Some of the attainable advantages made possible by design rationale capture include:

- cost savings realized from less time spent on repetitive decisions
- improvement of the quality of artifacts due to the incorporation of past decisions
- enabling designers to incorporate lessons learned from others
- supporting linkages between associated artifacts
- preventing loss of information
- providing a tracking system for decisions
- fostering accountability (Jarczyk, Loeffler, and Shipman, 1992, pp. 577-586).

Just as each project's design rationale will differ, the benefits afforded by the capture will vary from scenario to scenario.

C. DESIGN REUSE

Once captured, design rationale must be used in order to derive any benefits; the use of existing artifacts, of which design rationale is the principal artifact, is referred to as design reuse. Typically, only the reuse of artifacts such as code is attempted in reuse efforts. The availability of design rationale will greatly enhance the potential of artifacts not traditionally utilized in reuse efforts. By capturing the design rationale which lead to particular artifacts, information on the context in which the artifact was developed may be evaluated. Additionally, incorporation of design rationale may facilitate the "tailoring" of existing artifacts for reuse in current contexts.

Even more important than examining the artifacts is the ability to incorporate the process by which they were created. Now, instead of just asking the question, "What should be examined," designers are asking "Why was this done the way it was done." Typically there is a multitude of artifacts available which reflect the results of design decisions, such as user's manuals, design sketches, source code, etc., but there may be no tangible record of the deliberations which lead to design decisions. Effective reuse would encompass the ability to discern the design decisions from the design deliberations.

Once equipped with the ability to identify relevant material, the next step is the actual incorporation of this information into the process. The term, process, encompasses a broad range of activities in the design reuse realm. Designers may be modifying an existing system, designing a system to function in a similar environment, designing a system which performs the same function, or redesigning the total system from the ground up; all of these areas could profit from design rationale reuse.

III. COMMUNICATION

Communication is a natural element of every day life. From the time that we are first aware of our environment and continuing throughout our life, we are constantly utilizing and refining our ability to communicate. Each of us employs our own unique way of communicating. Understanding individual communication processes and how they combine to form group processes will enhance the understanding of design.

Designers or managers working in the field of software design are constantly wondering what can be done to improve the process. They examine all facets of the process from the beginning stages of design to the creation of artifacts which result. Throughout, one factor is always present and is key to all other functions, namely communication. This word, communication, encompasses many realms. While it is natural to think of communication as speech or the words we use to convey a thought, the act of communicating encompasses much more. The basic building block of communication is language. "Speech is verbal communication, while language is the body of formal rules governing the use of symbols and signs, be they lingual, vocal, verbal, gestural, or otherwise nonverbal" (Matson and Montagu, 1979, p. 174).

A. INDIVIDUAL SKILLS

As a child we begin to learn language skills from our parents and from the influences that surround us. We learn that "a human being... is never dependent on his own experience alone for his information" (Hayakawa, 1939, p. 10). As we grow older the influences to which we are subjected multiply. The most central of these becomes our formal education. Here we have the ability to learn from those outside our immediate proximity.

> Instead of remaining helpless because of the limitations of (ones) own experience and knowledge, instead of having to discover what others have already discovered, instead of exploring the false trails they explored and repeating their errors, (one) can go on from where they left off (Hayakawa, 1939, p. 10).

Formal education, although the cornerstone, is not one's sole source of experiential learning. A person's cultural and social backgrounds also influence communication assimilation skills. "People are more than a sum of parts...they have a set of values, goals, and beliefs about life and work" (Greenbaum and Kyng, 1991, p. 28) all of which are formed by the influences which surround them. These influences can have a direct bearing on the formulation of design rationale and subsequent reflection in the artifacts. For example, individuals, who from an early age have been fostered with a

strong morale responsibility from their parents, will manifest that belief as an adult in their work place by exhibiting strong ethical work behaviors. By understanding the influences by which design rationale are formulated we may be able to more fully utilize the artifacts of the design process.

B. GROUPS

Focusing on the area of design, we realize that by necessity, much work is done in groups. Newstorm and Pierce (1990, p. 105) stated:

> Some tasks are too demanding, too difficult, or too important to be performed by individuals. Because of their great potential for diverse perspectives, their combined breadth of member experiences, and the powerful support they can provide for decisions made, groups represent key building blocks for organizations.

There are certain dynamics involved when individuals come together to form a group. We will focus on three of these: cooperation, coordination, and integration. Cooperation represents the state of mutual support among individuals Coordination can be viewed as the existence of actions that are synchronized in some way to produce a common benefit. However, the mere presence of coordination does not necessarily result in a cooperative relationship among the

individuals involved. We will refer to integration by Aronoff and Baskin's definition as a higher level of organization that includes both cooperation and coordination to produce a molding of individuals and activities into a unified whole (Aronoff and Baskin, 1980, pp. 58-64).

Integration is not an automatic phenomenon, it is the result of interaction between group members over a period of time which culminates into distinct behavioristic patterns. Group members will develop expectations concerning one another's behavior in these patterns. In doing so, they will come to identify one another as members of the same social entity (Aronoff and Baskin, 1980, pp. 58-64). This coming together as a unique entity, with certain expectations, norms and behaviors is what we see as the formulation of group culture.

Much has been written about group culture. Webber's (1969, p. 10) surrealistic view of culture is described as:

> We are immersed in a sea. It is warm, comfortable, supportive, and protecting. Most of us float below the surface; some bob about, catching glimpses of land from time to time; a few emerge from the water entirely. The sea is our culture.

In contrast, Kroeber and Kluckhohn (1963, p. 357) offer a more concrete description:

> Culture consists of patterns, explicit and implicit, of and for behavior acquired and transmitted by symbols, constituting the distinctive achievement of human groups, including their embodiments in artifacts; the essential core of culture consists of traditional (i.e., historically derived and selected) ideas and especially their attached values; culture systems may, on the one hand, be considered as products of action, on the other as conditioning elements of further action.

Regardless of the view one chooses to believe, the fact that group culture exists is undeniable. This culture defines the who, what, where, why, and how of the group (Aronoff and Baskin, 1980, p. 58).

As previously stated, culture arises from individuals interacting as a group. To illustrate the group process, one can think of the process as a play. Within this group, each individual will play a certain role. Assignment of each member into their individual roles is delineated by the group's culture (Cooper and Payne, 1981, pp. 59-71). It is also the culture that "scripts" the roles of each the actors or participants against a common backdrop.

Group culture can be viewed as the ordinary behavior or the norm under which a group functions. It is assumed, expected, and what is normally performed (Greenbaum and Kyng, 1991, pp. 121-126). The formulation of this day-to-day

routine results from the internal dynamics exhibited by the group members as well as the group's position in the formal and informal organizational structure (Cooper and Payne, 1981, p. 96).

Remembering that groups are comprised of individuals who must constantly communicate with one another, it is important to state a major barrier to productive interpersonal communications. This major barrier within all interpersonal communications is the psychological aspect of language. These subtle nuances of emotion are added to the common semantics of language combined to portray unique messages. Therefore, in order to understand the message being conveyed the receiver must strive to listen to what the sender means rather than what he says (Fellows, 1964, p. 28).

One may ask how these subtleties of culture relate to design rationale. As previously stated, artifacts are the most common outcome of design rationale; through the examination of these artifacts, researchers and designers endeavor to understand the design rationale that produced such an outcome. However, "meanings do not...exist in artifacts, symbols, or practices...they are assigned...by people who perceive and interpret their content and context" (Smircich, 1983, pp. 160-172). To interpret accurately the perception of the group which created the artifact, one must understand that

group's culture. As Greenbaum and Kyng (1991, p. 125) pointed out that "cultural manifestations (artifacts) are easy to obtain but difficult to interpret because they are ambiguous and may hold multiple meanings and understandings." Again, it is the understanding of group culture that will facilitate and allow this interpretation.

In attempting to understand group culture, one will normally try to find a common reference. This reference may easily be one's personal background or knowledge. So, instead of looking for the peculiarities, one is really looking for similarities. As Geertz (1973, p. 14) wrote "understanding a people's culture exposes their normalness without reducing their particularity."

To understand the group process, one must first understand the underlying communications between individuals or groups in the design process. The complex merging of individual and group communications is the foundation of group culture. It is upon this culture that the everyday processes are formed. In the context of design rationale, the type of group process of primary importance is group deliberations; as explained in Chapter II, design rationale is the outcome of group deliberations.

When designing or choosing a tool, architecture, or system to capture design rationale, one must remain mindful of the

communications aspect of the design process. Simply capturing the design rationale, or the end product of the process may not be as helpful as capturing all the communications elements surrounding the design rationale which lead to that end product. Current research will be elaborated upon in Chapter IV which highlights the need for capturing communications as well as design rationale.

IV. DESIGN RATIONALE CAPTURE

Methodology can be defined as "a collection of procedures, techniques, tools, and documentation aids which will help the systems developers in their efforts to implement a new information system" (Avison and Fitzgerald, 1988, p.4). A mechanism to capture critical aspects of the design process is an important choice to be made by the designers; how they choose to capture the execution and any reuse of the design can be just as crucial as what they design.

Mechanisms providing a clear record of the employment of a design methodology is a critical component of an effective design process. The objective of such a mechanism is to accurately and systematically record all phases of the design process, to assist in the tracking of costs and time requirements, foster the development of a user friendly, well documented product, and allow for adaptability to change not only during the development stages of the design but also throughout the life cycle of the system.

A variety of systems have been developed in recent years that aim at supporting the capture and reuse of design process information. Some of these tools provided are based on information models that represent design rationale

information, while others focus on capturing group communication aspects of the design process.

We review some of the most advanced systems discussed in the literature in order to understand the salient components of a comprehensive model for representing design rationale information, as well as identifying potential technologies that could be used to support its capture and use. Descriptions of the research methodology employed in the development of these systems is also provided to illustrated the various methodologies that need to be employed to fully understand the group design process.

A. TANG

John C. Tang's dissertation work on *Listing, Drawing and Gesturing in Design: A Study of the Use of Shared Workspaces by Design Teams* in the Department of Mechanical Engineering at Stanford University investigates whether "the needs of a group using a tool collaboratively, are different from those of an individual user, and these differences should be reflected in the design of the technology" (Tang, 1991, p.143). In order to accurately assess these needs, Tang conducted a series of video taped sessions of groups designing various products. The outcome of these sessions would enable the understanding

of the design process and identification of opportunities to support group design practices.

In any research project, researchers must first identify a methodology upon which to base their research. Analytical methods commonly utilized to collect data include experimental, protocol and interaction analysis.

In the experimental method, studies are conducted in a controlled environment such as a laboratory setting. Normally two groups are set up for the experiment: a control group and the experimental group. The experimental group is subjected to preset conditions which are under study. Analysis of the differences between the two groups allows researchers to determine the effects of the preset factors and to judge whether they were a result of the experiment or some other existing condition.

When the factors being studied involve group interactions "the validity of an experimental approach involving human activity is reliant upon the ability to:

- manipulate the behavior of the participants to construct different conditions,
- accurately measure the outcome of each condition for comparison,
- collect a sufficient sample size of repeatable activity to validate the results and compensate for individual variation" (Tang, 1989, p. 46).

"Team design work is a rich activity that does not lend itself to conventional experimental methods of systematic study" (Tang, 1989, p. 45).

The basic methodology employed in protocol analysis is the verbalization of thought patterns by the participant during some predefined task. Subjects are asked to think aloud while working on the task. These deliberations are captured by the researchers in the form of video or audiotape for later analysis.

Protocol analysis is appropriate when studying an individual, but its appropriateness in groups is questionable. Asking participants to verbalize all their thoughts could disrupt the group process and stifle creativity.

Interactive analysis examines the details of human interaction in groups and is the main thrust of Tang's research involving the use of gestures in the design process.

Tang's interaction analysis approach emulates a qualitative analysis method commonly used in social sciences. In Tang's interaction analysis, subjects were unobtrusively observed in a natural working environment. Video and audio tapes of the sessions, combined with the actual artifacts produced, are utilized to capture all interactive processes between group members.

Groups of three or four individuals are involved in conceptual design tasks. The sessions last between one to one and one-half hours, with the actual duration at the discretion of the group. Two video cameras are set up to observe the group's problem solving process. These stationary cameras are mounted on a wall in a non-intrusive manner such as not to distract the group dynamic process. The use of two passive cameras is chosen over a manned camera because it is believed to be less distracting to the participants. One camera captures the group interactions from a distance, while the other is centered towards the middle of the working space to capture hand gestures, drawing, and listing. To complement the recording process and later transcription, an audio tape is made of each session.

Each session is initiated with a briefing given to all participants by the experimenter. This brief covered the basic task which each group would perform. Afterwards the experimenter observes the process in an adjoining room via the recording equipment. Following each working session an informal debrief, facilitated by the experimenter, is conducted to capture the initial feelings about the group experience from each participant's point of view.

Initial analysis of the data is accomplished by studying the video tapes and involves:

- becoming familiar with the data,
- developing a working representation of the data for analysis,
- abstracting observations from the data (Tang, 1989, p. 56)

Familiarity is accomplished by making a transcript of the video and audio tapes. Utilization of the *NoteCards*, a hypertext system, enables the experimenter to catalogue the data in a user friendly manner. *Notecards* is a transcription system which can be used to break down the data (or conversation) into segments. These segments contain an idea or simple activity which is typically less than one minute in length and involve three to seven conversational turns by the participants. The cards are then linked by theme and ordered in a chronological manner. Tang utilized *Notecards* to catalogue ideas and identify reoccurring activities.

Interactions of the group, as well as the artifacts produced, comprise the workspace activity which is subdivided into three main dimensions: the first being the composition and the capabilities of the workspace itself; secondly, the kind of task being performed; and finally, the working dynamics of the group. Composition describes the materials

available to the participants such as drawing paper, chalk boards, etc. while the capability of the particular composition describes the utilization level of the material itself. For instance, the utilization of a large piece of drawing paper which provides a drawing surface useable by more than one participant would differ from that of a single sheet of paper upon which only an individual could focus.

The category, kind of tasks, represents the nature of the task being performed such as graphics, textual, or interactive skills. Length of time and stages of development are major components of the activity which are categorized by the kind of task.

The last category, working dynamics, examines interactions and behavioral patterns displayed by group members during the performance of the task.

The framework that captured the dimensions of workspace activity "lays out relationships between *actions* that occur in the workspace, and *functions* that are accomplished through those actions" (Tang, 1989, p. 67). Accordingly, all group activities are broken down into the components of *actions* and *functions*. *Actions* are described by either List, Draw, or Gesture activities while *functions* are described by Store Information, Express Ideas, or Mediate Interaction. A matrix, designed with *functions* being the row indicator and *actions*,

serves as the column indicator. Within the matrix, four aspects of an interaction are highlighted: conventional view, gestural expression, expressing ideas and mediating interaction. This framework enables the experimenter to describe interaction in overlapping methods which expands the conventional views previously believed to solely describe group work. "This exercise not only led to a deeper familiarity with the data, but also helped focus the analysis on trends that became apparent in the data" (Tang, 1991, p. 149).

Categorization of data is accomplished through solicitation of various perspectives. By incorporating the inputs gathered from multiple sources such as the participants, engineers, software designers, etc. It is hoped that the design team will form a global view of the data. This solicitation takes place in meetings in which the participants vary.

Studying group processes enables the identification and capture of many workspace activities that mediate the groups collaborative actions which are not normally captured in the artifacts.

B. DEDAL

Researchers from Stanford University and NASA Ames Research Center have developed an Electronic Design Notebook as a part of a design reuse assistance product called Dedal. The goal of their research is to be able to capture design information during the conceptual design phase in engineering design for later reuse.

Researchers are concerned with the capture of conceptual design information in the least intrusive way. The use of multimedia, such as video, audio, text and graphics aids, has been selected for this reason. Because of the overwhelming amount of information that could be produced, the team needed a way to categorize the data to facilitate easy retrieval as needed. An indexing system utilizing a query based language has been developed for this purpose. The main goals of the language are to provide ease of use and to reduce the amount of redundancy in the data collection and retrieval process.

The development of the Electronic Design Notebook is the first step in achieving a reuse system. The Notebook which could be carried by the designer has been created to assist in the capture of technical and graphic documents as well as information about the designer's thinking process. A smart work surface that consists of a word processor, graphics

interface, and indexing capabilities is provided at the discretion of the designer.

Organization of the data to facilitate reuse involves the transformation of data into a format usable by a query based retrieval system. This data was indexed by the designer during the capture stage using key ideas (tagged words) which would later be utilized in its retrieval. A major problem with the Notebook was the complexity of the data retrieval process.

Dedal is the progressive extension of the Notebook. Dedal, like many other generic design rationale tools, is a "system that uses (a language) to:

- enable the description of the design record and content,
- help engineers formulate questions (concerning the project under design),
- select appropriate records in answer to a question" (Baudin, et al., 1992, p. 702).

One of the main strengths of Dedal is its language and indexing system. It has been developed to be primarily used during mechanical engineering design processes and its applicability in other design fields such as software engineering has not yet been explored.

Dedal was designed to "describe the content and form of design records such as meeting summaries, pages of an electronic notebook, technical reports and videotaped conversations between an expert designer and a novice" (Baudin, et al., 1992, p. 702). Once this information is captured, it is the responsibility of Dedal to format it in such a way as to facilitate reuse. To ensure this, the language of Dedal had to represent primitives to encompass the content of design record, including the design process, and form of the design activities. The specifications document is an example of the "content" or purpose of a record. The design process feature captures all discussions which take place regardless of whether the option being discussed was accepted or rejected. The "form" of the design information is made up of the level of detail of that individual piece of information such as global view and medium in which it is portrayed such as textual or graphical representation.

Retrieval of indexed data is facilitated by heuristics used in querying the database for answers to specific questions posed by the designers and engineers. These heuristics are required to address "two issues that (made) the retrieval of design information especially difficult:

(1) the (design) concepts evolve over time and

(2) design concepts are closely interrelated" (Baudin, Baya, and Gevins, unpublished, pp. 6-7).

Completed queries are matched with record descriptions in order to determine whether relationships exist between the queries and the concepts in the model.

Indexing patterns are formatted in the following categories: information, topic, subject, level of detail, and medium, in the form of information about topic (T) regarding subject (S) with level of detail (L) using medium (M). The following are the options under each category:

Topics	Subject-Class	Media	Levels of Detail
strategy	assembly	text	conceptual
reference	component	picture	configur-ation
description	connection	schematic	detailed
location	feature	photo	
function	design-concept	video	
operation		table	
dependency		equation	

The above vocabulary may be expanded to accommodate specific aspects associated with any design projects.

Once the designer has formulated the question or query for the data retrieval, the indexing patterns are used as the basis for the retrieval system. If a match is made, Dedal returns a set of references to the user. If any of the

references are available on-line, then the user could access them immediately. If a match is not found, then Dedal goes through a set of heuristics to "loosen" the match.

Heuristics are categorized into two classes: retrieval and ordering. The retrieval heuristics select the indexing patterns which relate to the query being used. The ordering heuristics define how to order the references being selected.

The retrieval heuristics are further divided into two types: proximity and causal relations. Proximity heuristics look for areas of related information and assumes needed data will be located near these regions. Causal heuristics look for dependencies among the attributes of the requested data.

If the heuristic match is found to be reliable then index acquisition can be utilized to create new indices. "The new indices created are expected to increase the precision and recall of the retrieval." This strategy can be termed question-based acquisition. Effectiveness of new indices are rated by reusability, relevance, and context independence in future retrievals.

The developers of Dedal have presented a method for the intelligent indexing and retrieval of design rationale information. They have utilized the talents of knowledge engineers and mechanical engineers to help with the initial indexing of vast amounts of data. From this process they

continue to develop a language which enabled the retrieval of the information. By utilizing custom heuristics, the retrieval mechanism is continually refined.

The Dedal tool, as presented by the researchers, is expandable, and therefore, should allow possible implementation in other fields and applications. We believe Dedal may be incorporated into and enhance other capture mechanisms by tailoring the language to meet specific applications addressed in those tools.

C. CONVERSATIONBUILDER

ConversationBuilder was designed by the Delta Group as part of ongoing research at the Human-Computer Interaction Laboratory at the University of Illinois. Their approach to the design process is centered around the communications aspect of group interactivity, namely conversations.

The creators of ConversationBuilder observe that humans function in distinct thinking modes, some of which require conscious thought and others do not. Conversations arise when one wants to convey part of their subconscious thoughts to another or wants to change another's thought patterns. Specifically, "a conversation is a structured sequence of linguistic acts which:

- serves a medium for communication,
- facilitates recovery from breakdown,
- provides synchronization between the participants,
- is the mechanism for manipulating the specification" (Carroll, 1992, pp. 17-18).

However, the subject or function around which the conversation centers commonly changes during the conversation itself.

The goal of ConversationBuilder is to accurately and systematically capture all the nuances of the conversations shared between designers. In order to accomplish this task, multiple conversations would take place at the same time, or in parallel and some mechanism would have to be provided to capture and later categorize these conversations. As the number of designers involved in any one conversation increases, so does the possibility for an increase in the number of subjects into which they simultaneously delve. This adds yet another complexity to the design of ConversationBuilder.

The actual software components which comprise ConversationBuilder are the Message Bus, Conversation Processor, and User Interface Suite. A conversation is started with a message sent from the user to the Message Bus. The message is comprised of a header, made up of a tag and domain address. An example of a tag is "status" of a message

such as "available" or "busy." The domain address may combine one or more recipients. "ConversationBuilder operates by components sending messages to each other over the Message Bus. The Conversation Processor operates as a central transaction processor. Users request transactions by activating display objects in their user interfaces" (Carroll, 1992, p. 28).

The Delta group has developed a bank transaction center scenario ɔ illustrate the functionality of ConversationBuilder. The cycle begins with "Wait for Customer." At this point a user or "customer" initiates a request to the system or "teller". A request consists of deposits, withdrawals, and inquiries from the customers on a particular account. Conversations between the customer and teller revolve around granting and denying these requests. Only one request and one customer is allowed within the system at one time. Customers not being served must wait for the "Wait for Customer" status to indicate that the teller is free to take their request. Each request results in the creation of a transaction record which is stored in a database. A hypertext module consisting of nodes or transaction records and links or relations between records is also created. Chronology of the conversation is maintained by the system to ensure accuracy of the account transactions. Distinct pieces

of dialog from the conversations are referred to as utterances while the entire conversation is an instance of a particular class or protocol. Action spaces result from the input of utterances to the Conversation Processor. "The parts of a conversation are:

- an instance of the protocol class
- a set of participants
- an action space for each participant" (Carroll, 1992, p. 168).

ConversationBuilder was intended to be used to capture conversations in the design process. ConversationBuilder did not prove to be a useable system as stated by Carroll (1992, p. 262):

> (It) supports design in a poor to mediocre way. It cannot be considered a "good" design support system in the sense of providing a complete and design support environment. The system has a number of inadequacies that prevent the generation of smoothly functioning design support applications.

One particular inadequacy centered around its inability to support commitments in conversation. In other words, it could not provide a strong mechanism to track whether taskings were

completed by those individuals who indicated a "commitment" to them.

Aside from ConversationBuilder's shortcomings, Fischer asserts his belief that his research in the formulation of ConversationBuilder provides an excellent theoretical basis for further research in this area. Since conversations are the center of all deliberations, the study of ConversationBuilder has great relevance to the capture and reuse of design rationale.

D. NETWORK-HYDRA

As the size and complexity of design projects grow, the number of people involved in the project correspondingly increases. Although the focus of many individuals' work may be the same project, each may work at different locations or at different points in time. In both cases some form of collaboration would benefit the design process even though face-to-face collaboration may prove difficult or perhaps not feasible. The designers of NETWORK-HYDRA recognized the need to support collaboration among members of design teams when direct communications between them were impossible or impractical.

NETWORK-HYDRA provides a mechanism that would allow an individual to work independently, yet be alerted when any

aspect of their work had some impact on others' designs. By alerting the designer of conflicts or correlations of their work with data in the system, NETWORK-HYDRA would allow the designers to assess the impact of their work on the overall project, thus enriching the individual designers' understanding of how other designers work in the project is relevant to their own. By functioning in this manner, the system "could effectively create virtual cooperation between all designers who ever worked on the project" (Fischer, et al., 1992, p. 285).

The work conducted by researchers at the University of Colorado, University of California, Irvine, and GMD, Darmstadt, Germany have three major goals in the development of NETWORK-HYDRA: integration of collaboration efforts; integration of construction, reuse, and specifications; and support of the creation of design rationale.

The largest problem that the researchers face in the capture of design rationale information is motivating the designers to impart it. Unless the individual designer feels there is something to gain from the capture, it is unlikely that the individual would be willing to aid in the process. With this in mind, the research team has developed the idea of creating a "seed" which contained a skeleton of a design environment to which the designers could input information.

Vast amounts of information are commonly required in a single project and it is unlikely that one person would have the expertise to utilize or even the need for all the data available. However, having such data readily available to designers as they require it could facilitate the design activity. Two key functionalities of the NETWORK-HYDRA system are the facilities to allow designers to retrieve the material relevant to their area of design and to alert them to problems associated with their individual task which may conflict with others' work. The information which would allow these functionalities requires the system to integrate collaborative work as well as individual efforts.

The design process as seen by the research team is comprised of two states: action and reflection. They view the designer as typically working in a nonreflective manner until a breakdown or problem occurred. At this time, the designer's process changes to a reflective state in order to repair the breakdown. Once a "fix" is achieved, either good or bad, the process returns to the nonreflective state. These are referred to as construction (action) and argumentation (reflection). To integrate the two, NETWORK-HYDRA alerts the designer when a breakdown occurs.

To capture these states, a hypermedia system is utilized because it allows for a multiplicity of connections and offers

the availability of media other than text. The gIBIS model is the basis for the preliminary model incorporating the researchers' own language, PHI (Procedural Hierarchy of Issues). PHI focuses of the dependency relationships between issues and how interdependencies affect the system as a whole.

The following example on the construction of a network illustrates the functionalities of the NETWORK-HYDRA architecture. Network systems are normally evolutionary in nature; this is due to changing configuration requirements necessitated by varying connection needs and changes in hardware and software development. With most network changes commonly occurring over a long-term basis, managers of the system need to be aware of past decisions and problems associated with any changes as well as current requirements. As turnover of personnel can occur during this time, a system to capture this information is needed so the system is not dependent solely on the knowledge of its users and/or designers.

A domain knowledge base which consists of design parts, rules and discussions was constructed from an existing network system, is used as the initial "seed" prototype for the hypermedia system.

Since design projects tend to evolve over time, the system is designed to provide end-user modifiability. "To support

evolution on a continual basis, the people experiencing the breakdowns are in the best position to do something about them" (Fischer, et al., 1992, p. 296). The seed, therefore, is formulated in a manner which allows it to evolve over time. As modifications occur because of breakdowns in the system, the seed could accurately reflect these changes.

The NETWORK-HYDRA system architecture is comprised of the following components:

- construction kit
- argumentative hypermedia
- specification component
- catalog
- simulation component.

The construction kit provides a palette for the designers to utilize during the design stage. With the rapid changes in technology in mind, the designers have built end-user modifiable palettes to allow for maximum utilization of available collaborative tools.

The argumentative hypertext incorporates PHI as well as other multimedia environments. The primary purpose of the hypertext is to provide the designers with the requisite information needed to repair a breakdown and continue with the

design process. Hypertext is used to represent detailed information on breakdown; possible issues, answers, and argumentation; and the design rationale of others in order to repair breakdowns.

The specification component allows designers to input system requirements and constraints associated with the task to "tailor its information structures by filtering out argumentation, critics, and catalog examples that are not relevant to the specific task" (Fischer, et al., 1992, p. 304). As the project evolves, changes to requirements are incorporated thus refining the specifications component. "In collaborative design, specifications serve to help coordinate the work of group members by providing a common framework in which to operate" (Fischer, et al., 1992, p. 304). Again, it is important to remember that changes made by individual designers affect the entire system and designers may not possess the knowledge about the entire system which would allow them to accurately predict the effect of those changes.

A catalog of design artifacts is readily available to designers. This facility provided a mechanism through which inclusions, deletions, and modifications could be made. This catalog also serves as the storage for designs, design rationale, and specifications for later reuse.

The last component facilitates simulation of scenarios in which "what if" conditions can be imposed.

Tools provided by NETWORK-HYDRA include: a construction analyzer to provide a critiquing system used during breakdown; a catalog explorer for use in searches of the system; and an argumentation illustrator which provides examples to be used by designers to promote understanding of the information presented.

Design rationale and reuse are key ingredients to all projects in all stages of development. This tool allows for the use of design rationale throughout the system's life cycle. Additionally, NETWORK-HYDRA reduces redundancy and maximizes the use of knowledge and skills of designers by allowing easy access to group memory.

E. gIBIS

The gIBIS tool was developed as part of the Design Journal project at Microelectronic Computertechnology Corporation (MCC). Design Journal is a hypertext tool which was designed to support system design processes. The developers viewed design journals as traditional and nontraditional documents and aspects, both of which could be supported by their tool. Traditional aspects include specifications, requirements, and design documents. Nontraditional aspects include components

or activities which are not normally archived as part of the design process such as interviews, scenarios, notes, sketches, design decisions and rationale, and design constraints. In addition to supporting both aspects, gIBIS also aimed at supporting the "upstream informal processes" which commonly surround deliberations encompassed in the formulation of design rationale.

There are two aims in the research which led to the gIBIS' design:

- understanding the internal structure of design decisions and their dependencies
- addressing interface problems associated with indexing and retrieval of vast amounts of informal data (Begeman and Conklin, 1988, p. 304).

The gIBIS tool provides graphical support for Horst Rittel's Issue-Based Information Systems (IBIS) method as illustrated in Figure 1:

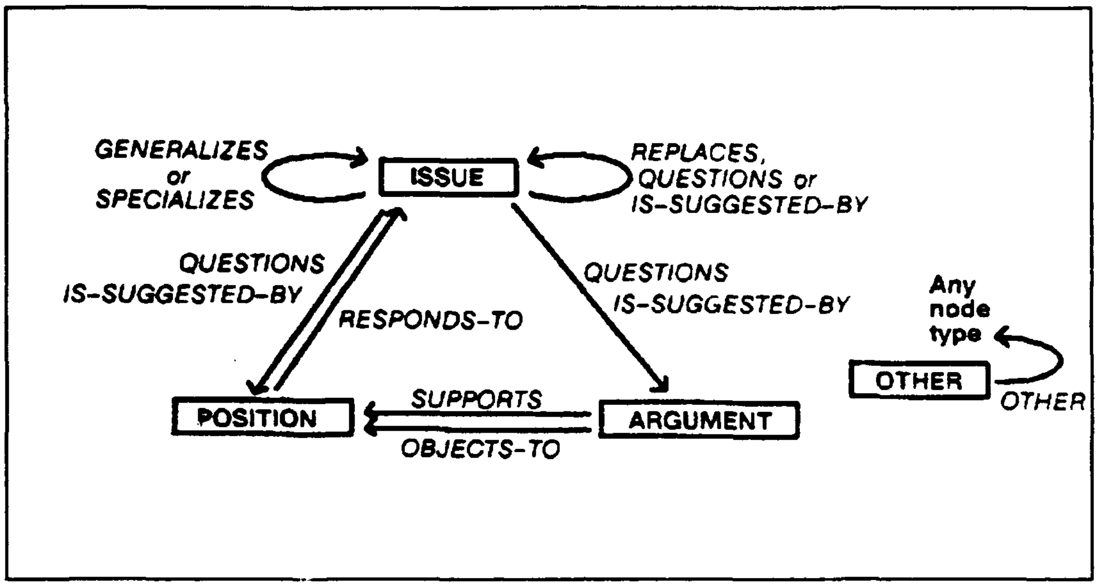

Figure 1 IBIS Model
Source Begeman and Conklin, 1988, p. 305.

The IBIS method was based upon the belief that the design process is basically a conversation between stakeholders, each of which contributes their concerns and expertise toward the resolution of design issues. All deliberations, whether they are in the form of problems, questions or concerns can be viewed as an *issue*. The addressing or resolution of issues is what Rittel viewed as the design process. Rittel summarizes the method by stating "the concept of IBIS rests on the model of problem solving by cooperatives as an argumentative process" (Kunz and Rittel, 1970, p. 1).

The IBIS method centers around the articulation of key

issues by the stakeholders. Each *issue* may have multiple *positions*. A *position* is a statement or belief which serves to advocate the issue. There may be one or more *arguments* which either support or object to a *position*. Each *issue* may be the root of a tree whose children are *positions*, which then may be parents for *arguments*. All of the nodes are connected by *links* which state the particular association of the two nodes being connected. For example, "supports" or "objects-to" links connect *arguments* to *positions*. In employing this method "... the goal of the discussion is for each of the stakeholders to try to understand the specific elements of each others' proposals, and perhaps to persuade others of one's viewpoint" (Begeman and Conklin, 1988, p. 305).

In addition *issues*, *positions*, and *arguments*, gIBIS also incorporates an *other* node. The purpose of this node is to allow the user to escape the theme upon which current work is centered and provide the capability to express and store unrelated information. An additional node called *external* is available for the storing of artifacts such as documents, sketches, and code. Extending the original IBIS model with nodes, gIBIS attempts to support both the design process and computer-mediated teamwork.

The user interface provided in the gIBIS tool consists of four tiled windows:

- graphical browser
- structured index
- control panel
- inspection window

As the name implies, the Graphical Browser provides a visual representation of the nodes and associated links. While working in this window, the user is afforded a global view of the project in the lower corner of the screen, while the central part displays a working model of the area of interest.

In the Node Index Window the user is shown a hierarchical view of the nodes of the current IBIS network. This offers the user an additional method to select nodes for more in depth examination.

The Control Panel provides functionality through the manipulation of buttons such as "next," "help," and "done." Each button offers a pull up menu and description of the function provided.

The Inspection Window offers a detailed description of the selected node and its links.

The system also offers two additional functionalities through the Tool Configuration Window and the Query Control and Help Window. These windows assist the user in setting parameters and searching for nodes through the manipulation of query based questions.

In addition to designing the functionalities previously mentioned, the design of gIBIS strove to address several other goals such as maximum reliability, support of multiple concurrent users, reasonably good performance, and implementation utilizing limited resources.

Users who function both individually and in groups have reported that gIBIS proves to be a useful tool (Begeman and Conklin, 1988, p. 323). For the individual, support in focusing thinking on difficult issues was aided by the IBIS framework. In the group realm, conversations were supported by the enforcement of a strict framework for discussions.

Even though the tool has proven to be beneficial, users did identify the following shortcomings:

- no specific nodes are available for the incorporation of goals and requirements
- no available facility for providing support for choosing among the various positions of an issue
- no method to link artifacts to specific areas within the gIBIS tool to facilitate the decision process (Begeman and Conklin, 1988, p. 324).

An inherent problem of IBIS identified by the developers of gIBIS was "segmentation." Because many conversations about design, especially in the early stages, are of the brainstorming type, identifying well defined *issues* may be difficult. Individuals may express thoughts which are vague, confusing or incomplete and labeling each of these as an individual *issue*, when they are in reality part of the same *issue* may be nonproductive. This type of behavior may lead to premature decisions. If in their enthusiasm for employing the tool, designers are not aware of this potential problem, they may not fully explore or identify the subject around which the *issue* is centered before they specify it. This phenomenon of premature labeling may cause information about the *issue* to be fragmented or spread into areas other than the *issue* which designers originally intended.

Although the availability of the *external* node offered a means to label artifacts, there was no specific functionality such as a menu choice to link the artifacts to particular nodes; this separation limited the full use of the tool to support all design deliberations.

F. REMAP

REpresentation and MAintenance of Process knowledge (REMAP) is a conceptual model designed to represent design

rationale and deliberations during decision making by providing a method to capture the entire process. REMAP includes the IBIS method to model argumentation processes.

REMAP's central goal is the capture of the entire history of the design process during all phases of the life cycle. During the life cycle, numerous artifacts are created, each with an accompanying set of documentation. An important component typically missing from this documentation is the rationale for the development of the artifact.

The REMAP project is specifically concerned with capturing rationale during the early stages of the systems development process, namely requirements engineering, because well defined requirements are critical to the development of high quality software. Recent research also suggests that reusability at the requirements stage is more productive than at the coding level. Availability of rationale will greatly enhance such reuse.

Numerous studies have highlighted the importance of capturing design rationale for the following reasons:

- multi-person teams involve communication and coordination between members
- long-term projects usually involve changing personnel and requirements which result in miscommunication and loss of information

- critical errors can result from lost data during decision making deliberations

- misinterpretation and misunderstanding occur in large projects over time involving different participants at different phases of work (Dhar and Ramesh, 1992, pp. 498-499).

The capture and reuse of design rationale is especially pertinent for large, complex projects.

> As these projects involve often large and complex problems, creation of design solutions involves knowledge that spans several areas....Since no single designer possesses all the knowledge required to produce a solution, a team of several members is typically involved in a design task (Dhar and Ramesh, 1992, p. 499).

REMAP includes the IBIS framework discussed in the earlier section involving concept types and relationships:

- *Issue*: a problem, concern or question

- *Position*: a solution which responds to an issue. Note that positions are not mutually exclusive

- *Argument*: statement that supports or objects to positions

Additionally, REMAP incorporates:

- *Requirement*: represent goals or objectives of the design problem

- *Design Object*: artifact which satisfies a requirement

- *Decision*: result of deliberations phase concerning issues discussed

- *Assumption*: idea taken to be true concerning an argument

- *Constraint*: restriction, limit, or regulation placed on the design object by a decision (Dhar and Ramesh, 1992, pp. 499-500).

Design efforts entail both individual efforts involving independent work and group problem solving resolving issues previously examined by the individuals. The REMAP model was derived based on an empirical study of individual and groups of experienced systems analysts involved in a requirements engineering task.

Two types of experiments were conducted. The first involved individual "think aloud" exercises in which the participants were engaged in a problem solving design task. The second involved the use of transcriptions from group meetings for requirements engineering. The experiment required the participants "to clearly articulate decisions made and reasons for making such decisions" (Dhar and Ramesh, 1992, p. 500). The REMAP conceptual model resulted from the analysis of this data. IBIS is the fundamental building block used to formulate the model. Additions to IBIS enable tailoring the model to the systems design content. Figure 2 illustrates the REMAP model:

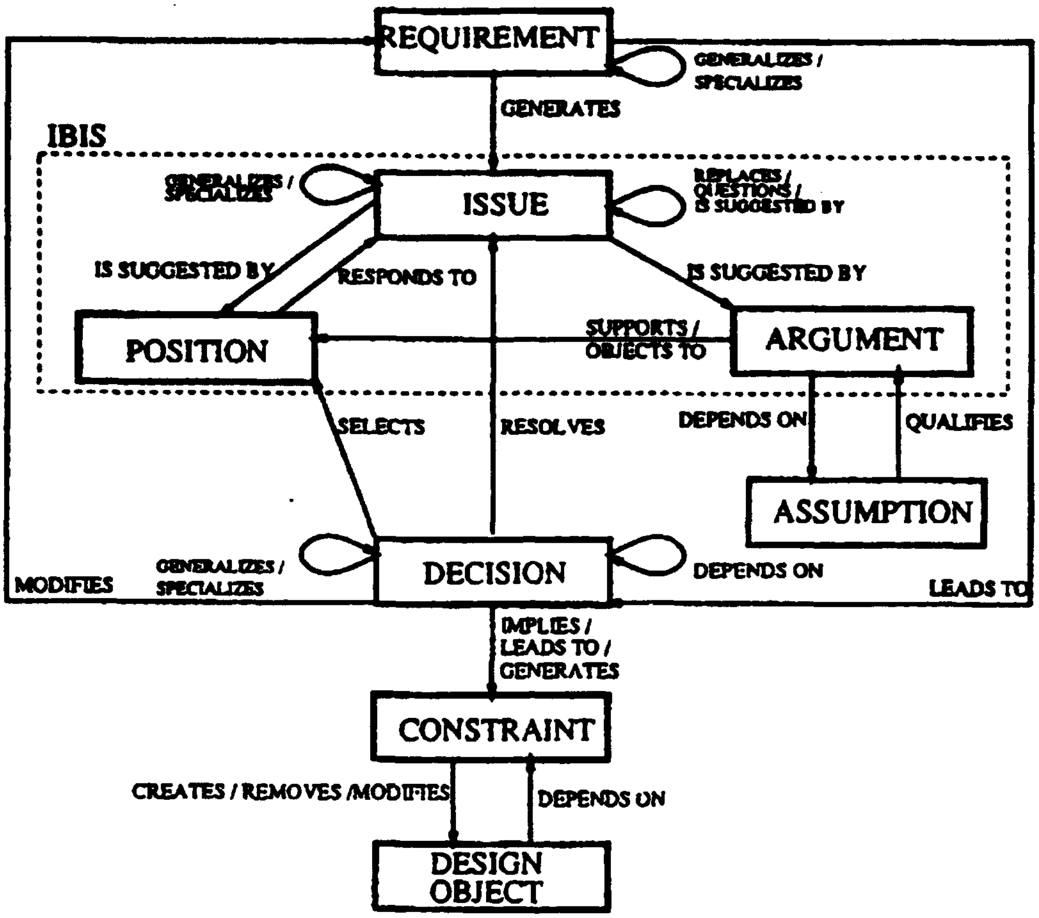

Figure 2 REMAP Model
Source Dhar and Ramesh, 1992, p. 501.

According to REMAP, the initial *Requirements* are set during the early stages of the design process. They represent the goal or objective to be met by the designers in achieving the *Design Object* or artifact. *Issues* are generated as deliberations are conducted concerning the stated *Requirements*

and also as individuals present problems that need to be resolved before proceeding on with the design. "Initial requirements get refined, modified, and elaborated during the deliberation process..." (Dhar and Ramesh, 1992, p. 501). *Issues* are formulated and refined in a hierarchical manner.

Participants assume various *Positions* concerning these *Issues* which are supported by or objected to through the use of *Arguments*. *Assumptions* about a particular *Argument* are explicitly represented. Ultimately a *Decision* or set of *Decisions* is reached by the group concerning each *Issue* or set of *Issues*.

REMAP extends IBIS by incorporating the artifacts that are resultants of design deliberations. These artifacts or *Design Objects* are linked to the decision through constraints that are implied, generated, or led to by decisions resulting from the deliberations.

The prototype software incorporating the REMAP model is based on the software package called ConceptBase, which implements TELOS, a high-level conceptual modelling language. ConceptBase was selected for its client-server architecture that could support distributed design processes.

REMAP provides facilities for the construction, querying, and maintenance of structured knowledge bases which are the building blocks for capturing and storing of design rationale.

The REMAP tool supports interactive instantiation, querying, and modification of instances of REMAP objects. Interactive use of the tool facilitates the incremental acquisition of process knowledge or design rationale. In order to allow for a convenient traversal of the knowledge base created, a hypertext browsing capability is provided. REMAP provides the user capability to browse, display and edit existing design rationale objects at any phase of the design process.

REMAP, in contrast to gIBIS, provides primitives such as *Requirements, Decisions, Assumptions, Constraints*, and *Design Objects*. Further, besides providing an extended model, REMAP supports active reasoning with the design rationale knowledge.

V. ARCHITECTURE FOR DESIGN RATIONALE CAPTURE

A. OVERVIEW OF THE INFORMATION MODEL COMPONENTS

Primary outputs of the design process are design objects or artifacts. Current practices of documentation focus on representative outputs, ignoring the processes that lead to their creation. As discussed in Chapter IV, recent research has identified the importance of capturing the components of this process knowledge known as design rationale. In this research, our goal is to identify the components of an information model for design rationale and functionalities of mechanisms to support the capture and use of design rationale knowledge in design activities.

This chapter will discuss the following basic questions about design rationale:

(1) What information should be captured?

(2) What mechanisms should be provided to facilitate capture and use of the information?

We will answer these questions by first exploring what specific types of information should be captured and why they are relevant to design rationale and suggest examples of how

they can be incorporated into a design rationale information model designed for the capture and reuse of this information. Where applicable, we will cite specific tools or models which currently support such capture and use. Later, we will discuss the generic functionalities which we believe should be present in a design rationale management tool used to implement the design rationale information model.

B. INFORMATION MODEL COMPONENTS

The information model defines the content of the design rationale to be captured. Although there are numerous design rationale capture models, such as gIBIS, many address only limited aspects of design activities. There are useful mechanisms developed in other research that would aid in the capture and use of design rationale information, but they are not based on a comprehensive design rationale model. In this section we will expound not only on what should be captured by an information model, but why and provide examples of how that capture could take place; we believe the REMAP model which we explored in Chapter IV offers many of the fundamental building blocks necessary for such a design rationale information model. Instead of restating all the component descriptions, we will begin by suggesting viewing each of the following REMAP primitives: requirement, issue, position, argument,

assumption, decision, constraint, and design object, and various relationships among them. Our research has identified several other components that could be incorporated in an information model that includes the REMAP model. The following section describes these in detail:

1. Stakeholder - Characteristics

As projects become more complex in size and scope, design endeavors commonly involve groups of designers or stakeholders working together rather than single designers working independently. Thus, in order to understand design rationale resulting from a group process, one must first understand the group itself, the importance of which we expounded upon in Chapter III.

To gain additional insights into design rationale one should examine their sources. In a large design effort, each member of the group may have different interests. For example, one may be a project manager whose main objective is keeping the project on schedule, another may be a senior engineer who was brought into the project because of previous involvement on a similar project, yet another may be a customer representative whose primary goal is keeping costs to a minimum. All of these members, as stakeholders, will have unique perspectives and goals which could affect the design

rationale. Many insights into design rationale may be gained by the identification of the stakeholders who formulated them.

Examples of the stakeholder characteristics that could be included in such a model include:

a. Role

The method in which the individual stakeholders interact may also provide additional insights into design rationale. As we discussed in Chapter III, the phenomenon referred to as "group think" can allow authority figures to influence the group to make decisions which may not be technically sound. Illustrations of individual members' search for authority figures' approval or social acceptance is sometimes reflected in work habits. Therefore, an explicit link between design rationale and the role of every stakeholder who contributed to it is an important component of an information model.

b. Experience/Background

Additionally, each stakeholder will bring his or her own personal background and design experience. Naturally, experienced designers and novice designers will have vastly different work habits. For example, a novice designer would most likely go through the design process in a very methodical and step-by-step manner, whereas a more experienced designer would probably be less meticulous and would be more prone to

combine and delete minor steps and make assumptions that are not explicitly documented. Identification of the experience level and background of the stakeholder may improve the understanding and the potential for reuse of rationale.

Another example of an experience factor which may affect design rationale is the stakeholder's length of involvement in the project. Those who have a longer association with the project will have more knowledge about its idiosyncracies.

With changing project teams, the goals and priorities of the group may change. Being able to identify such changes can help identify possible sources of design rationale which embody the changes.

The people involved in single projects are not necessarily located in one area. Ready identification of location will provide clues as to what special considerations were made to enable non-collocated stakeholders to work in conjunction with one another.

The characteristics described above are only some examples of "background" information which could be captured as properties of the stakeholder. Additional varieties of characteristics such as their "stake" in the project could also be captured, based on the intended use of such information in design activities.

2. Gesture - Body Language, Drawing, Listing

Although design rationale may be explicitly reflected in textual artifacts, some of the communications in their formulation cannot be adequately preserved in a textual representation. In the Chapter IV, we discussed Tang's research which explored the importance of capturing the gestures that accompany human communications involved in design activities. These activities include body language, listing, and drawing. There is an old saying that "a picture is worth a thousand words" which holds much relevance in the use of multimedia to capture gestures. Simply having an observer take notes and transcribe them at a later date or having designers input textual details of design rationale does not capture the full essence of interactions. Further, such recording actions may interrupt the design process.

Passive video taping could be a non-intrusive capture mechanism that would reflect the multi-dimensional activities of gesturing because "we lack a ready descriptive vocabulary for bodily behavior which could be captured in notes (therefore) the looks, the body orientations, all of that is lost and probably not recoverable from memory" (Greenbaum and Kyng, 1991, p. 79). The mechanisms required to easily identify and retrieve this information are discussed later in this chapter.

3. Issue - Characteristics

There are several important characteristics of issues which need to be resolved during the design process which, if captured, would greatly enhance the usefulness of design rationale information. Examples of such characteristics of issues include:

a. Time stamp

Knowing how design rationale were formulated may provide a more in depth understanding of the rationale itself. The order in which design issues were introduced may at a cursory glance appear trivial, however, the sequence may be as important as the eventual rationale which was formulated. Being able to explore the existence of such sequencing and subsequent correlations would help in the understanding of the thought patterns employed by the designers.

There are current mechanisms which exist to capture such dynamics and allow replay of activities; an illustration is seen in REMAP which allows chronological or design dependency-directed replay of design rationale information. Examples of useful temporal information include a time stamp indicating when the issue was created or when it was resolved. An attempt to employ this type of information was detailed in the ConversationBuilder discussion in Chapter IV.

b. Status

Tracking project status can be accomplished by identifying "open" issues, or issues that have not been resolved. The gIBIS tool presently offers the ability to mark outstanding issues and has a query facility to retrieve them.

c. Prioritization and Resource Expended

Once the outstanding issues have been identified, additional capabilities should exist to allow for the prioritization of the tasks which would resolve open issues. Examples of factors which could influence the prioritization are criticality or complexity of an issue. For example, knowing ninety hours had been spent on issue "A" and twenty hours on issue "B" would provide a designer or project manager with more information than just knowing some time had been spent on issue "A," and some time had been spent on issue "B," and both are currently unresolved. Capture of design rationale related statistics may provide additional insight because factors like hours spent may be an indication of the complexity of the task which would help managers in prioritization of work. Other management functions such as tracking of issues to assess how to allocate resources such as man hours can also be aided by such information.

d. *Subject Area*

The typical project manager continually deals with outstanding activities, but having the capability to define the subject area of the issue will enable categorization or selective retrieval of unique issues or groups of issues. Such information could assist in gaining insight as to why a class of issues remains unresolved. The design rationale of the unresolved issues may also indicate trends in problems which hinder issue resolution or may function as indicators for upcoming or future problems if certain types of issues pose recurring difficulties. Functionalities that provide tracking of issues should possess a flagging mechanism that allows the user to identify, compute, and correlate statistics on outstanding issues by subject areas. NETWORK-HYDRA currently possesses such flagging mechanisms and allows for the identification of unresolved issues.

4. Project Dictionary

The design rationale information model should include a tailorable project dictionary because design objects and artifacts commonly possess project specific terms and language. Comprehension of these terms can be facilitated by selectively recalling definitions from the project dictionary. A tool which supports this functionality could use hypertext terms that allow the users to click on specific text and

automatically recall detailed definitions. As explained in Chapter IV, this functionality is available in the Dedal tool which could be built into a mechanism which implements the design rationale information model.

5. Constraint/Requirement - Source

Design may be viewed as a constraint satisfaction activity where some constraints are explicitly stated in the requirements and others arise from the refinement of the basic requirements through design activities. Besides explicit representation of the constraints, ability to trace where the constraints came from would help in design situations where constraints need to be relaxed.

6. Representation of All Alternatives

Issues are resolved by evaluating alternatives. It is not sufficient to capture only the alternatives chosen to resolve an issue because alternatives that are discarded for various reasons may become relevant in a changed context. As assumptions, constraints, or requirements change, the discarded alternatives may be preferred over the "chosen" one. In the absence of a complete record of various alternatives considered, resources must be dedicated to reformulating these from scratch. To phrase it in layman's terms: "the designers may be reinventing the wheel." A model that allows for the

identification and isolation of the alternatives such as REMAP primitives, *position* and *argument* may alleviate this problem.

7. Design Rationale and Artifact Linkages

Very often interest revolves around reusing the artifact rather than the rationale. Indicating clear linkages between design objects, or artifacts, and the design rationale will help in the understanding of the context in which the artifacts were created. Many available models for design rationale, such as gIBIS, capture only the design rationale but provide no discernable link to the artifacts produced using this design rationale. Ideally the user should be able to select a design object and accumulate all the design rationale used to create it. This information is especially important, in an evolutionary system design situation where changes will inevitably be made to the design object as the project progresses. Examples of such a linkages are the relationships between *design objects* and the decision *constraints* in REMAP.

C. GENERIC FUNCTIONALITIES

The design rationale information model components we have suggested would be the basis from which a tool to capture and use design rationale could be built. Any tool, in order to

provide maximum assistance to the user, would need to provide at least the following generic functionalities:

1. Semi-Structured Tailorable Environment

An important consideration in the development of a model for design rationale capture and use is the degree of structure that could be imposed with such a model. A totally unstructured capture of design rationale information will significantly affect the potential for its use. Simply videotaping the design activities that lead to the formulation of design rationale would be an example of such unstructured capture. On the other hand, videotaping can provide a non-intrusive means of capturing design information. Although a very structured information model may constrain the designer and may prove to be intrusive, the information captured may be in a more useable format.

We see the answer as a compromise between the two extremes, a semi-structured environment where a flexible design rationale information model could be easily implemented.

As stated by the design team of NETWORK-HYDRA:

> Perhaps the single most difficult problem in getting information into the various components of group memory is that of motivating designers to impart this information. Nowhere is this problem more difficult than in the input of design rationale (Fischer, et al., 1992, p.286).

Additionally, they state a possible solution to this problem is creating an environment in which the designers see the need or benefit of capturing design rationale as a part of the task of designing. A mechanism which provides for a semi-structured environment in the capture process would greatly facilitate the use of such a system. NETWORK-HYDRA has accomplished this by providing a template from which the designers are able to create their own design environment:

> An important principle for our approach is that designers are more likely to use and to add to group memory of design rationale if they do not have to create project rationale entirely from scratch (Fischer, et al., 1992, p. 286).

To employ the design rationale information model, we believe the semi-structured environment must at least allow for three basic conditions.

First, the model employed within the environment should emulate the natural aspects of design process such as deliberations, similar to the gIBIS tool, so that its use is viewed by stakeholders as conducive to the design cycle, yet it should remain tailorable so idiosyncracies of the stakeholders' interests can be captured. One such example of attempting to provide a structure which was a reasonable representation of design processes is ConversationBuilder.

The creators of the tool believe that most design activities center around conversations between designers, therefore they developed a tool which they believe could enable the capture of conversations. Second, ideally the environment for employment of the model should be non-intrusive so that it does not disrupt the work flow process. Last, by allowing tailoring of the model, the environment could provide assistance across a spectrum of design areas from mechanical construction scenarios to software design projects.

2. Information Capture

As the size of projects increases, the number of design rationale used to create artifacts skyrockets. Believing that any one architecture can formally capture all aspects of large design projects is unrealistic. The use of video taping offers the capability to capture entire sequences of group interactions. Video clips could be categorized under hypertext headings and attached to various nodes. At later times the categorized information could be filtered should it become pertinent. For example, a video clip could be made of a session where a particular assumption is being explored. Instead of textually detailing every interaction of the session, the video clip could be attached to the assumption and should further review of the assumption be necessary the

video clip would be available simply by clicking on a hypertext node.

The ability to recall sessions at a later time is important because material which is irrelevant under one set of requirements or constraints may become relevant as requirements are refined in the design life cycle or data that is trivial to one issue may be relevant in other domains.

3. Representation Language

The design rationale information model components capture the rationale about the refinement, elaboration and modification of initial requirements or constraints that eventually lead to design artifacts. In order to support the representation and reasoning with such rationale a fairly powerful representation language is needed.

The design rationale would be stored in a knowledge base; therefore, the language must provide facilities to construct, query, and maintain structured knowledge bases. The content of such knowledge bases would consist of interconnected information model components that are incrementally modified.

The language should be capable of representing the ontology of design rationale in terms of the suggested model components and should provide mechanisms to populate the design rationale information model with specific instances.

The language should also provide automatic inferencing to enable access to the design rationale which is implicit in the model and provide mechanisms to maintain integrity of the knowledge base (Dhar and Ramesh, 1992, pp. 502-503). Such a language could also provide a basis upon which a decision support system could be constructed to further assist the users in such tasks as assessing the feasibility of alternatives or analyzing how a change in one area may affect other areas.

4. Information Exchange

The exchange of information between individuals is a primary activity in large scale design. Easy exchange of information is a basic characteristic which should be supported. By sharing information, designers can gain insights which could potentially strengthen their design rationale. Electronic "whiteboards," shared editors, and virtual conference rooms allow such exchanges to take place faster and more efficiently.

5. Simultaneous Work

To avoid duplication of efforts and maximize design talents, simultaneous work between project personnel, regardless of their relative locations, should be supported. In addition to fostering information exchange, the methods mentioned in the previous subsection will also enable

simultaneous work at geographically distributed locations. This is made possible by users being able to simultaneously access a central knowledge base with browsing and modification capabilities. At the recent Groupware'93 conference and exhibition in San Jose, California, many tools which allowed asychronous and geographically distributed exchange of information were displayed.

6. Levels of Granularity

Reuse of a design rationale knowledge base requires that the user must be able to traverse through stored information with ease. The ability to browse through the contents of the knowledge base at different levels of granularity will greatly enhance the usefulness of the knowledge base. For example, a project manager may want to look at issues at the highest level in terms of hardware and software issues. He may not want to see how issues at this level are broken down into sub-issues for resolution. A designer, on the other hand, may be interested in detailed descriptions of one these nodes. By allowing users to choose which level of granularity they wish to view, information overload can be avoided. The Graphical Browser provided in the gIBIS tool is an example of such a hierarchical representation of data.

7. Color Coding of Model Primitives

As the network of design rationale information could explode into several hundreds or even thousands of nodes and links in a large scale project, easy visual identification of various types of nodes and links would provide valuable assistance for navigating through such a network. REMAP and NETWORK-HYDRA both incorporate color coding schemes and use of different shapes to clearly identify various primitives. The design rationale information model could also emulate such a color coding scheme.

8. Cataloging of Decisions

Similar decisions are made time and again over the life span of a project or even across similar projects, therefore, the ability to identify, track, and make inferences about such relationships should exist. An indexing system should allow the users to classify decisions by decision types. Such an indexing system will facilitate the development of a Decision Support System (DSS) to assess degrees of similarities between decisions, help identify when decisions conflict with one another or perhaps even illustrate otherwise inexplicit relationships between decisions. A DSS could also assist in understanding their relationships. An indexing system such as the one in Dedal could be

incorporated into a design rationale capture tool such as REMAP.

9. Decision Support Facilitation

In the context of group design, mechanisms that facilitate arriving at a group solution will be valuable. Multi-Criteria Decision Making (MCDM) methods may be employed to evaluate *positions* and *arguments* to arrive at a group solution.

VI. RECOMMENDATIONS AND CONCLUSIONS

Doing more with less in the shrinking budgets of both industry and within the Department of Defense necessitates the reevaluation of current and past practices. In this thesis, we have suggested one way to refocus the practices in the area of design by concentrating on the design process itself rather than the products of the process. Specifically, attention should be directed at the capture, understanding, and reuse of design rationale.

We have suggested the basic components which should be in a design rationale information model; discussed the importance of such components; explored examples of current technologies and models that exist to capture such components; and suggested the generic functionalities of a tool which could be used to employ the design rationale information model.

We believe this thesis contains the foundations upon which a specific design rationale information model can be built. By incorporating existing technologies, such as those presented here, we believe a semi-structured, non-intrusive architecture could be constructed to provide virtual communications between all the stakeholders who ever participated in any aspect of design deliberations. Providing

virtual communications would allow for effective and efficient capture and reuse of a plethora of design rationale and enable the users to actually do more with less.

LIST OF REFERENCES

Aronoff, C.E., and Baskin, O.W., *Interpersonal Communication In Organizations*, Goodyear Publishing Company, Inc., 1980.

Avison, D.E., and Fitzgerald, G. *Information Systems Development: Methodologies, Techniques and Tools*, Blackwell Scientific Publications, 1988.

Baudin, C., Baya, V., and Gevins, J., "Using Device Models to Facilitate the Retrieval of Multimedia Design Information," unpublished paper submitted to IJCAI-93.

Baudin, C., et al., "Dedal: Using Concepts To Index Engineering Design Information," Proceedings of the Meeting of the Cognitive Science Society, Bloomington, Indiana, 1992.

Begeman, M.L., and Conklin, J., "gIBIS: A Hypertext Tool for Exploratory Policy Discussion," *ACM Transactions on Office Information Systems*, v.6, pp. 303-331, October 1988.

Carroll, A.M., *ConversationBuilder: Building Blocks for Open Collaborative Systems*, Doctorate Thesis, University of Illinois, Urbana, Illinois, 1992.

Cooper, C.L., and Payne, R., *Groups at Work*, John Wiley & Sons, 1981.

Dhar, V., and Ramesh, B., "Supporting Systems Development by Capturing Deliberations During Requirements Engineering," *IEEE Transactions on Software Engineering*, v. 18, pp. 498-510, June 1992.

Endoso, J., "House Takes Tough Stance on DoD Budget," *Government Computer News*, p. 6, 20 June 1992.

Fellows, H.P., *The Art and Skill of Talking with People: A New Guide to Personal and Business Success*, Prentice-Hall, Inc., 1964.

Fischer, G., et al., "Supporting Indirect Collaborative Design With Integrated Knowledge-Based Design Environments," *Human-Computer Interaction*, v. 7, pp. 281-314, 1992.

Geertz, C., *The Interpretation of Cultures*, Basic Books, Inc., 1973.

Greenbaum, J., and Kyng, M., *Design At Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates, 1991.

Gruber, T., and Russell, D.M., "Beyond the Record and Replay Paradigm for Design Rationale Support," paper presented at the AAAI'92 Workshop on Design Rationale Capture and Use, San Jose, California, 15 July 1992.

Hayakawa, S.I., *Language in Thought and Action*, 4th ed., Harcourt Brace Jovanovich, Inc., 1978.

Jarczyk, A., Loeffler, P., and Shipman, F., "Design Rationale for Software Engineering: A Survey," Hawaii International Conference on Systems Sciences, 25th, pp. 577-586, 1992

Kellogg, W.A., Maass, S., and Rosson, M.B., "The Designer As User: Building Requirements for Design Tools from Design Practices," *Communications of the ACM*, v. 31, pp. 1288-1298, November 1988.

Kluckhohn, C., and Kroeber, A.L., *Culture: A Critical Review of Concepts and Definitions*, Knopf, 1963.

Kunz, W., and Rittel, H.W., "Issues As Elements of Information Systems," working paper No. 131, July 1970.

Matson, F., and Montagu, A., *The Human Connection*, McGraw-Hill Book Company, 1979.

Newstrom, J.W., and Pierce, J.L., *Windows Into Organizations*, American Management Association, 1990.

Smircich, L., *Beyond Method. Strategies for social research*, Sage, 1983.

Tang, J.C., "Findings from observational studies of collaborative work," *International Journal of Man-Machine Studies*, v. 34, pp. 143-160, March 1991.

Tang, J.C., *Listing, Drawing, and Gesturing in Design: A Study of the Use of Shared Workspaces by Design Teams*, Doctorate Thesis, Stanford University, Palo Alto, California, 1989.

Webber, R.A., *Culture and Management*, Richard D. Irwin, Inc., 1969.

DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3.	B. Ramesh, Code AS/RA Naval Postgraduate School Monterey, California 93943-5002	5
4.	Kishore Sengupta, Code AS/SE Naval Postgraduate School Monterey, California 93943-5002	1
5.	Lieutenant Debra A. Morris 17115 Graves Court Huntersville, North Carolina 28078	4
6.	Lieutenant Tina M. Paco UNCMAC PSC 303 Box 54 APO AP 96204-0054	4